TECHNICAL DATASHEET

JEWEL™

Jewel Low-Cost 13.56MHz Radio Frequency Identification (RFID) Read/Write IC (ISO 14443A Compatible)

Part Number: IRT101DA(A)

Confidential Material - NOT TO BE RELEASED WITHOUT NDA

INNOVISION RESEARCH & TECHNOLOGY PLC

Revision 1.3a June 2004

Table of Contents

Datasheet Revision History

Revision	Date	Page(s)	Description
1.0	08/03		Product Specification
1.1a	12/03		Text amended
1.2a	01/04		Text amended
1.3a	06/04		Text amended. Section 16 added.

UK Head Office

Innovision Research & Technology plc
Ash Court
23 Rose Street
Wokingham
Berkshire RG40 1XS, UK

Telephone: +44 (0) 118 979 2000
Fax: +44 (0) 118 979 1500

Email: enquiries@innovision-group.com
Web: www.innovision-group.com

1. Description

The Jewel integrated circuit (part number IRT101DA) has been developed by Innovision Research & Technology plc to address low-cost Radio Frequency Identification (RFID) tagging applications working to the ISO/IEC14443A standard.

The Jewel IC is a two terminal device designed to be connected to a loop antenna to produce a passive RFID tag operating in the standard unlicensed 13.56MHz frequency band.

The on-chip tuning capacitor is metal mask selectable to suit a variety of antenna coil dimensions and characteristics.

The read/write data in the tag memory is EEPROM-based, allowing individual blocks to be locked into read only operation by contactless command. Once locked, the process is irreversible.

The Jewel is based on a physical EEPROM array size of 120 bytes.

Passive operation means that no battery is required because the Jewel RFID tag gathers its operational energy from the interrogation field generated from the reader unit.

2. Features

- Jewel IC for use in Proximity Integrated Circuit Card (PICC) applications
- Designed to be compatible with ISO/IEC14443 parts 2 and 3
- ISO/IEC14443 type A modulation scheme
- Passive RFID tag operating in the unlicensed 13.56MHz band
- Read and Write (R/W) operation
- One Time Programmable (OTP) & Write Once Read Many (WORM) operation
- Typical operating range up to 10cm depending on tag/reader antenna coil sizes and orientation relative to the reader unit
- Fast data communication rate 106 kbit/s
- Collision detection with immediate halt function provides protection for data during a write operation in the situation where there are multiple tags in the reader field
- Anti-tear protection available (achievable by virtue of memory capacity)
- Fast byte write speed
- Data communications are protected by 16-bit CRC security
- EEPROM based user read/write memory area organised as 12 blocks of 8-bytes
- 7-bytes of Unique Identification (UID) number for use in data authentication/anti-cloning
- 96-bytes of user read/write memory
- 6-bytes of OTP memory
- 2-bytes of metal maskable ROM for fixed operator/scheme/product specific data
- All memory areas are individually one time lockable by RFID command to prevent further modification of data and to produce read only functionality

3. Benefits

- Low-cost due to ultra small die size
- Blocks of memory can be utilised as shadow areas for anti-tear protection measures
- Initial “Request and Answer” communication cycle between the Proximity Coupler Device (PCD) and the Jewel based Proximity Integrated Circuit Card (PICC) follows the ISO/IEC14443A part 3 standard
- Operates with typical existing ISO/IEC14443 readers after software modification only
- Memory size and capacity is scalable for custom designs
- Suitable for operation with wide variety of antenna coil size, form factor and construction methods
- Wire-bond, flip-chip and module die attachment methods
- Metal mask options for selection of non-standard tuning capacitances
- Low-power requirement
- High-security – 16-bit CRC and other features retain integrity
- Fast read command (RALL) saves testing time
- Metal mask custom ID option
- Two bond pad die attachment

4. Specification

Physical/Environmental

- Sawn die size 0.59 x 0.59mm
- Standard 150µm thickness
- L1, L2 pad passivation opening ≥ 80µm
- 2 terminal IC for conventional wire bond or flip chip attachment
- Operating temperature range: –23°C to +50°C
- Non-operational data retention (i.e. storage temperature) range: –23°C to +50°C

Memory Map

- 16-bits (2-bytes) of metal mask product identification option
- 56-bits (7-bytes) of Unique Identification (UID) number
- 768-bits (96-bytes) of user read/write memory
- User Read/Write memory arranged as 12 blocks of 8-bytes
- Each 8-byte user read/write block is individually lockable by RFID command
- For systems working on 16-byte blocks, the pairs of 8-byte blocks can be written to and locked together by the reader
- 48-bits (6-bytes) of One Time Programmable (OTP) area

Security

- 7-byte Unique Identification (UID) number is programmed and locked during manufacture
- Further blocks can be programmed with application specific data and then locked to provide tamper-proof contents
- OTP bits can be used for indelible one direction counters
- CRC protection on command and data communications to retain integrity
- All blocks, and hence all logical pages, have a one-time lock capability
- The Jewel IC uses a “Digital Certificate” or “Seal” based on the unalterable and unique identification number to authenticate and provide an appropriate level of security.

General

- On-chip tuning capacitance designed for nominal 13.56MHz operation using ISO ‘credit card’ size coil of between 6 to 8 turns
- Metal mask options for selection of non standard tuning capacitance values allow for a wide variety of antenna coil size, form factor and construction methods
- Fast write speed <6.5mS per byte
- “Read all” command for fast read access of complete memory contents
- Data retention >5 years
- Write operations >10,000 cycles

5. C-tune

C-tune is the poly-poly capacitance connected across the device pads L1 & L2 (expected use: to tune a coil connected across L1 & L2 to a frequency near to 13.56MHz). C-tune can be metal-mask selected to any value up to the maximum in block size increments.

- C-tune max setting = 27pF
- C-tune block resolution = 1pF
- C-tune standard = 20pF nominal
- C-tune parasitic = 2pF typical

6. Physical Memory Map

The 120-byte EEPROM array is arranged as 15 blocks of 8-bytes each. Each block is separately lockable. Additionally, there is a 2-byte Header ROM (metal-mask data selected) – these 2-bytes are called HR0 and HR1.

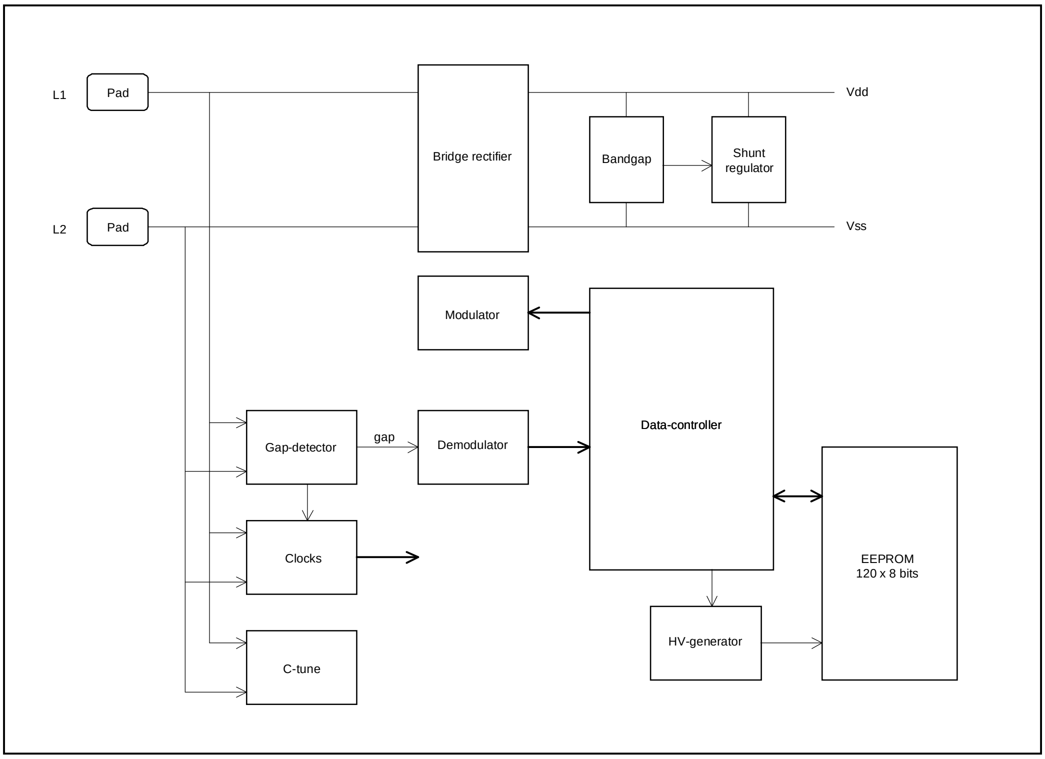
HR0	HR1
01	00

EEPROM Memory Map									
Block	Byte-0	Byte-1	Byte-2	Byte-3	Byte-4	Byte-5	Byte-6	Byte-7	Lockable
0	UID-0	UID-1	UID-2	UID-3	UID-4	UID-5	UID-6		Locked
1	Data0	Data1	Data2	Data3	Data4	Data5	Data6	Data7	Yes
2	Data8	Data9	Data10	Data11	Data12	Data13	Data14	Data15	Yes
3	Data16	Data17	Data18	Data19	Data20	Data21	Data22	Data23	Yes
4	Data24	Data25	Data26	Data27	Data28	Data29	Data30	Data31	Yes
5	Data32	Data33	Data34	Data35	Data36	Data37	Data38	Data39	Yes
6	Data40	Data41	Data42	Data43	Data44	Data45	Data46	Data47	Yes
7	Data48	Data49	Data50	Data51	Data52	Data53	Data54	Data55	Yes
8	Data56	Data57	Data58	Data59	Data60	Data61	Data62	Data63	Yes
9	Data64	Data65	Data66	Data67	Data68	Data69	Data70	Data71	Yes
A	Data72	Data73	Data74	Data75	Data76	Data77	Data78	Data79	Yes
B	Data80	Data81	Data82	Data83	Data84	Data85	Data86	Data87	Yes
C	Data88	Data89	Data90	Data91	Data92	Data93	Data94	Data95	Yes
D									Locked
E	LOCK-0	LOCK-1	OTP-0	OTP-1	OTP-2	OTP-3	OTP-4	OTP-5	OTP

	Reserved for internal use
	User Block Lock & Status
	OTP bits

Block usage	
Block 0	7 Bytes Unique ID
Blocks 1 – C	All 96 data bytes are available to the user as Read/Write
Block D	Least significant 4-bytes are reserved for internal use. By arrangement the most significant 4-bytes can be made available as custom fixed ROM data.
Block E	Used for OTP (One Time Programmable) bits once the device is initialised. The least significant 2-bytes are used to store the block-lock status. The most significant 6-bytes are available as OTP bits for the user.

7. Block Diagram



8. Device Operation

The Jewel device will operate in ISO/IEC14443 type A mode – Proprietary branch at 'Check ATQA' (ISO/IEC14443-3 section 6.4.1).

Collision detection is provided so that the reader knows if there is more than one tag in its field. This collision detection makes use of the 4 least significant bytes of the UID (Unique Identification number).

On power-up, the Jewel device remains 'silent' unless a REQA or WUPA command is received.

If the Jewel device fails to decode some incoming data, and is in a state waiting for the missed data, then if a new command frame is received, the device resets to the "received command" state.

This situation could occur, for example, if the Jewel device was a tag at the edge of range – then the reader/coupler could re-try the command.

9. Demodulation (data sent to the device)

According to the Type-A variants in the ISO/IEC14443-2:2001(E) & ISO/IEC14443-3:2001(E) specifications. Type-A uses 100% carrier modulation.

10. Modulation (data sent from the device)

Modulation is achieved by varying the impedance of the tag as 'seen' by the coil – according to the Type-A variant in the ISO/IEC14443-2:2001(E) & ISO/IEC14443-3:2001(E) specifications. Reference to BS ISO/IEC10373-6:2001.

11. Frame Formats (data sent to the device)

Each seven or eight bit data set is sent to the device in a separate frame (i.e. a command sequence will usually consist of several frames).

S = start of frame
E = end of frame

Command (7 bits)

“Short Frame” as specified in ISO14443 part 3, type A

	LSB						MSB	
S	b1	b2	b3	b4	b5	b6	b7	**E**

Transmitted first

Operand or Data (8 bits)

Innovision-proprietary 8-bit version of the ISO Short Frame

	LSB							MSB	
S	b1	b2	b3	b4	b5	b6	b7	b8	**E**

Transmitted first

After power-up the device will only recognise the ‘S’ as a valid ‘start of command sequence’ and a command will only be considered valid if there are 7-bits between ‘S’ and ‘E’.

12. Frame Formats (data sent from the device)

Data output from the device is sent as a single frame. For each command response the 8-bit data bytes (together with each byte’s parity bit) are concatenated into one frame.

The frame format is the “Standard Frame” as specified in ISO/IEC14443 part 3, type A.

S = ‘start of frame’ followed by one or more bytes (LS bit first in each byte). Each byte is followed by a P (parity bit) where the number of 1’s is odd in (b1 to b8, P).
E = end of frame (after last byte’s parity bit).

				1st Byte							2nd Byte				nth Byte			
S	b1	b2	b3	b4	b5	b6	b7	b8	P	b1	....	b8	P	b1	....	b8	P	**E**

13. Commands

13.1 Command List

Command		Operation (7 bits) MSB b7	b6	b5	b4	b3	b2	LSB b1	Comment (all commands are independent)
REQA	26	0	1	0	0	1	1	0	Request Command, type A
WUPA	52	1	0	1	0	0	1	0	Wake-up, type A
RID	78	1	1	1	1	0	0	0	Read ID – Use to read the metal-mask ROM and UID0-3 from block 0
RALL	00	0	0	0	0	0	0	0	Read All (all bytes)
READ	01	0	0	0	0	0	0	1	Read (a single byte)
WRITE-E	53	1	0	1	0	0	1	1	Write-with-erase (a single byte)
WRITE-NE	1A	0	0	1	1	0	1	0	Write-no-erase (a single byte)

The device will ignore any other Operation bit patterns than those of the eight commands above (i.e. no internal operation and no data response).

Apart from REQA & WUPA, 8-bit operand frames follow all commands. The Address operand has a particular format and is shown in this table:

Address operand Block = select one of blocks 0 – E Byte = select one of byte 0 – 7							
MSB							LSB
b8	b7	b6	b5	b4	b3	b2	b1
	Block				Byte		
X	MSB			LSB	MSB		LSB

Apart from REQA & WUPA, a 2-byte CRC is included in each part of the command and response sequence. If the CRC received by the device does not match the one generated as data arrives, then the device will halt the operation and move to 'Command End' status waiting for the next command. The CRC is the 16-bit version as specified under CRC-CCITT – for definition see ISO/IEC 14443-3:2001(E) Annex B: CRC_B. Single-bit error detection is expected to be detected at 99.998%. CRC is to be calculated on all data bits including the header bytes HR0 & HR1. However, start, end, parity (and the CRC bits themselves) are not included within the CRC calculation. See annex for example.

Additional security for RALL, READ, WRITE-E & WRITE-NE is provided by 'addressing' the device by sending its UID back to it, as part of the command. If the UIDs do not match, then the device will halt the operation and move to 'Command End' status waiting for the next command.

Command-response summary table

Greyed-out frames are dummy frames – their data content is 'don't care' as they are only required for the device internal logic

Command	Command (Multiple frames to the device)								Response (Single frame from the device)						
REQA	REQA								ATQA0	ATQA1					
WUPA	WUPA								ATQA0	ATQA1					
RID	RID	ADD	DAT	UID0	UID1	UID2	UID3	CRC1	CRC2	HR0	HR1	UID0	…	UID3	CRC1	CRC2
RALL	RALL	ADD	DAT	UID0	UID1	UID2	UID3	CRC1	CRC2	HR0	HR1	UID0	…	DATe	CRC1	CRC2
READ	READ	ADD	DAT	UID0	UID1	UID2	UID3	CRC1	CRC2	ADD	DAT	CRC1	CRC2			
WRITE-E	WRITE-E	ADD	DAT	UID0	UID1	UID2	UID3	CRC1	CRC2	ADD	DAT	CRC1	CRC2			
WRITE-NE	WRITE-NE	ADD	DAT	UID0	UID1	UID2	UID3	CRC1	CRC2	ADD	DAT	CRC1	CRC2			

Command-response timing table			
Command	Command bytes	Response bytes	Total command-response times (ms)
REQA	1	2	0.4
WUPA	1	2	0.4
RID	9	8	1.9
RALL	9	124	11.7
READ	9	4	1.6
WRITE-E (5ms prog-time)	9	4	6.6
WRITE-NE (2.5ms prog-time)	9	4	4.1

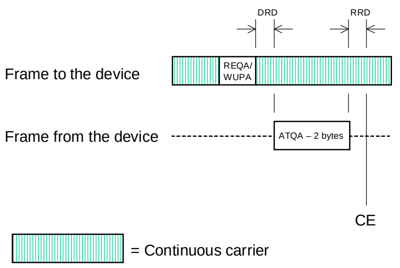
Command timing: (bit period; B = 9.4µs)

1. Frames to the device: S = 1B; data = 8B; E = 2B; inter-frame gap = 2B
2. Command-response delay = 10B
3. Frame from the device: S=1B; E = 2B; data = 9B per byte

13.2 Command Details

Timing & description definitions (used in the command sequence descriptions below)		
Name	**Description**	**Specification**
RRDD	Reader-Reader Data Delay	≥ 9.44µs
DRD	Device Response Delay	ISO/IEC 14443-3:2001(E), section 6.1.2
RRD	Reader Response Delay	ISO/IEC 14443-3:2001(E), section 6.1.3
CE	Command End	Idle = device does nothing until further commands are received
UID-echo	The four LS UID bytes from block 0 (LS-byte first)	
Programme-time-E	Wait while device programmes data into EE array	5ms minimum
Programme-time-NE	Wait while device programmes data into EE array	2.5ms minimum

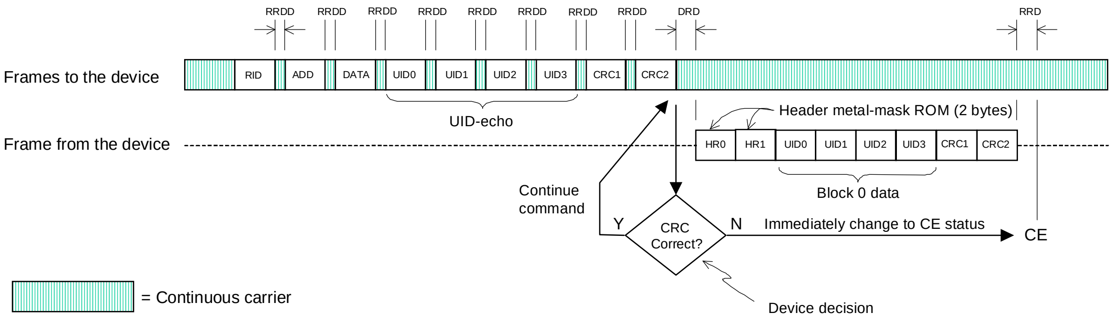
13.2.1 REQA or WUPA



The ATQA bits are to be metal-mask selectable (Option 2 – see Section 14), and for the first instance all 16 bits are to be set to zero. ISO/IEC 14443-3:2001(E), section 6.4.2.1 refers. REQA (or WUPA) is used to conform to ISO/IEC 14443-3:2001(E), section 6.4.1. Together with the appropriate response, ATQA, it will lead the reader/coupler down the "proprietary anti-collision" branch. In this case the 'anti-collision' method will be that the reader/coupler detects a collision – during reading of the device ID – and warns the user/system to act accordingly.

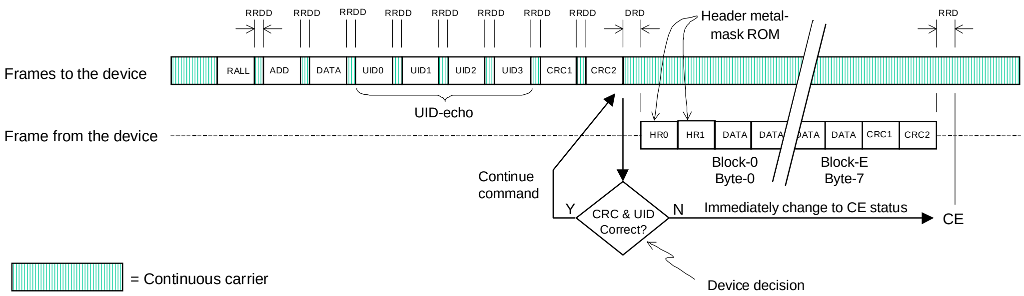
Use of the REQA-ATQA (or WUPA-ATQA) response should make it relatively easy for reader/coupler manufacturers to cater for using this device, because this conforms to the Part-3 specification.

13.2.2 Read Identification (RID)



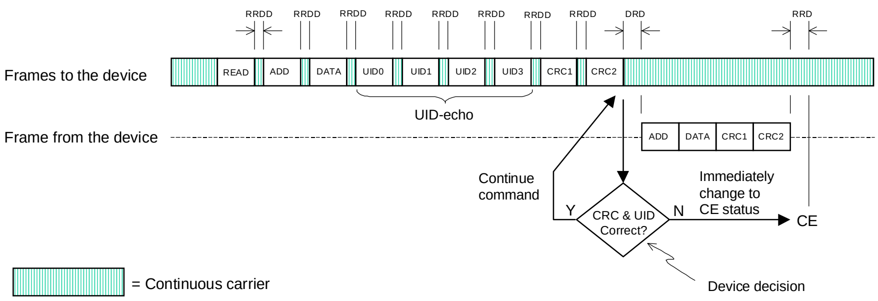
RID (Read Identification) reads the metal-mask header bytes and the four LS UID bytes from Block-0. The command frame, then Address, data-byte, UID-echo & CRC data frames are sent to the device. However, the address, data & UID-echo bytes can be set to any value – RID needs these bytes to be sent, but will still operate whatever their value. If the CRC is correct, the read operation is carried out.

13.2.3 Read ALL Pages (RALL)



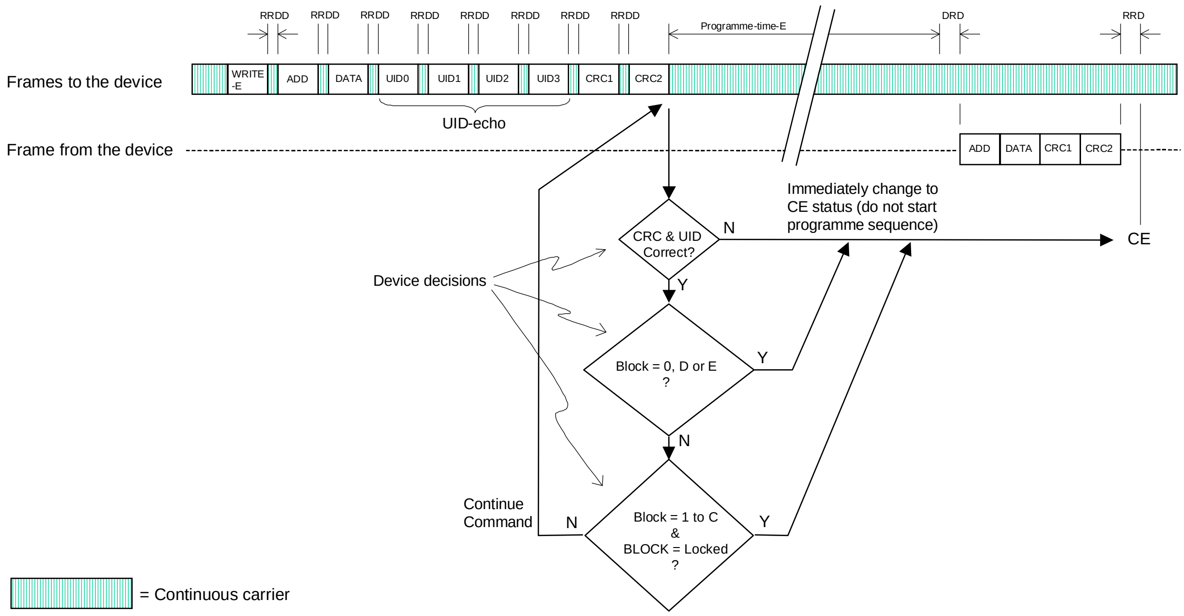
RALL (Read ALL pages) reads-out the whole device. The command frame, then Address, data-byte, UID-echo & CRC data frames are sent to the device. However, the Address & data-bytes can be set to any value – RALL needs these bytes to be sent, but will still operate whatever their value. If the CRC and UID are correct, the 2-byte Header ROM (metal-mask), followed by all the EE memory, and then the frame CRC are then output to the reader/coupler.

13.2.4 READ



The READ command relates to an individual EE memory byte – the byte address (Block number and Byte number) is sent with the command. The command frame, then Address, data-byte, UID-echo and CRC data frames are sent to the device. However, the data-byte can be set to any value – READ needs this byte to be sent, but will still operate whatever its value. If the CRC and UID are correct the memory byte is then read from EE memory. The Address, followed by the data-byte itself and the frame CRC bytes are then output to the reader/coupler.

13.2.5 Write-Erase (WRITE-E)



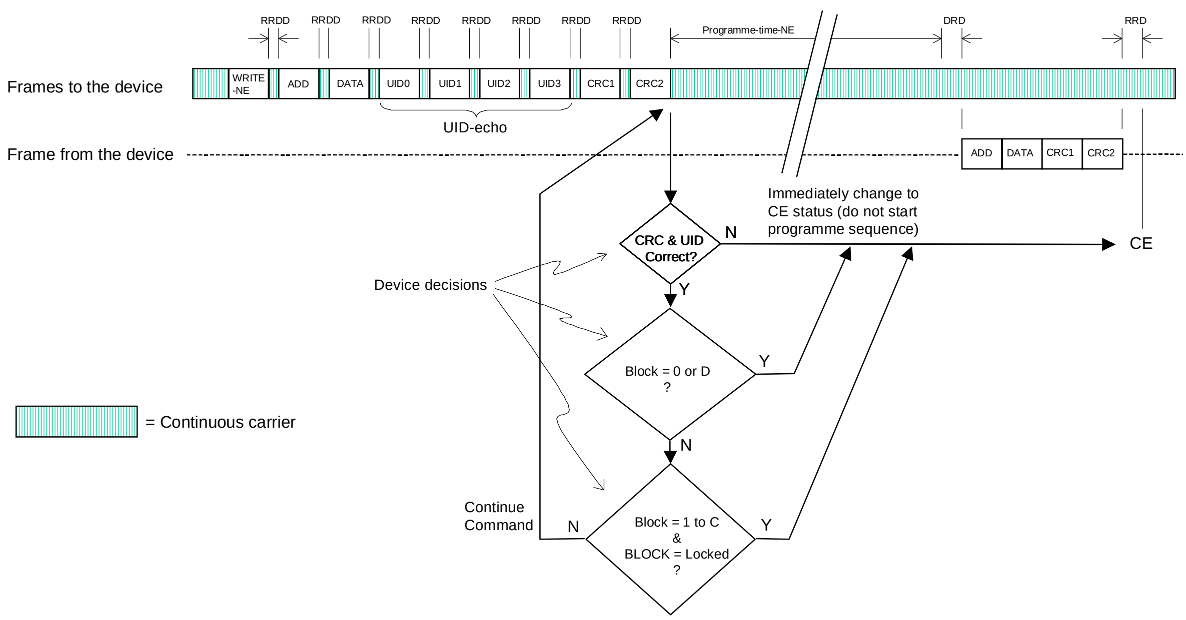
The WRITE-E (Write–erase) command relates to an individual EE memory byte – the byte address (Block number and Byte number) is sent with the command. This command does the 'normal' erase-write cycle (i.e. it erases the target byte before writing the new data).

If any of BLOCK-0 to BLOCK-D is locked then WRITE-E is barred from those blocks. Additionally, WRITE-E is barred from Blocks 0, D or E.

The command frame, then Address, data-byte, UID-echo and CRC data frames are sent to the device. If the CRC and UID are correct, (and WRITE-E is not barred), the EE memory write-cycle is carried out. The byte is then read back from the EE memory. The Address, followed by the data byte itself and the frame CRC bytes are then output to the reader/coupler.

If WRITE-E is barred, the erase-write cycle is skipped – no write operation occurs – and without waiting the programme-time, the device will enter CE status waiting for a new command.

13.2.6 Write-No-Erase (WRITE-NE)



The WRITE-NE (Write-no-erase) command relates to an individual EE memory byte – the byte address (Block number and Byte number) is sent with the command. This command does not erase the target byte before writing the new data, and its programme time is half that of the 'normal' write command (WRITE-E). Bits can be set but not reset (i.e. data bits previously set to a '1' can not be reset to a '0').

WRITE-NE is available for three main purposes:

- Lock – to set the 'lock bit' for a block (see 'Locking' below).
- OTP – to set One-Time-Programmable bits (bytes 2 – 7 in Block-E). Between one and eight OTP bits can be set with a single WRITE-NE command.
- Fast-write - to reduce overall time to write data to memory blocks for the first time.

If any of BLOCK-1 to BLOCK-C is locked then WRITE-E is barred from that block.

WRITE-NE is not barred from BLOCK-E to allow setting of lock and OTP bits.

The command frame, then Address, data-byte, UID-echo and CRC data frames are sent to the device. If the CRC and UID are correct (and WRITE-NE is not barred), the EE memory write-cycle is carried out. The byte is then read back from the EE memory. The Address, followed by the data byte itself and the frame CRC bytes are then output to the reader/coupler.

If WRITE-NE is barred, the write-no-erase cycle is skipped – no write operation occurs – and without waiting the programme-time, the device will enter CE status waiting for a new command.

13.2.7 Locking

All 12 blocks (1 to C) are separately lockable. When a block's 'lock-bit' is set to a 1, that block becomes irreversibly frozen as 'read-only'.

The lock-bits are stored in the two LS-bytes of BLOCK-E:

LOCK-1 (Byte-1 of BLOCK-E)								LOCK-0 (Byte-0 of BLOCK-E)							
b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
Not used internally	1	1	1 = BLOCK-C locked	1 = BLOCK-B locked	1 = BLOCK-A locked	1 = BLOCK-9 locked	1 = BLOCK-8 locked	1 = BLOCK-7 locked	1 = BLOCK-6 locked	1 = BLOCK-5 locked	1 = BLOCK-4 locked	1 = BLOCK-3 locked	1 = BLOCK-2 locked	1 = BLOCK-1 locked	1

During normal device use, the command WRITE-NE is used to set individual lock-bits. A single use of WRITE-NE can be used to set between one and eight lock-bits

14. Metal-Mask Selections

Option	Function	Section	Bits	Method	Comment
1	Header (0x0001)*	7	16	ROM	2-byte header data (HR0 & HR1)
2	ATQA (0x0C00)	8.2.1	16	ROM	16-bit response to REQA & WUPA commands
4	Tuning cap (20pF)	1.4	**	Separate bits	User to choose capacitor value between 0 – 27pF in 1pF steps

* 0x0000 denotes Engineering sample

15. Application Note

LoCoST or Limited Use mass-transit contactless smartcard ticket according to the anticipated Small Memory Card Model from the UK Integrated Transport Smartcard Organisation (ITSO)

The following tables show the suggested data mapping for storage of ITSO v2.1 defined ticket products within the Jewel IC for use in LoCoST and Limited Use tickets in a typical contactless smartcard mass transit scheme.

EEPROM Memory Map									
Block	Byte-0	Byte-1	Byte-2	Byte-3	Byte-4	Byte-5	Byte-6	Byte-7	Lockable
0	UID-0	UID-1	UID-2	UID-3	UID-4	UID-5	UID-6		Locked
1									Yes
2	ITSO-s	ITSO-s	ITSO-s	ITSO-s	ITSO-s	ITSO-s	ITSO-s	ITSO-s	Yes
3	ITSO-s	ITSO-s	ITSO-s	ITSO-s	ITSO-s	ITSO-s	ITSO-s	ITSO-s	Yes
4	ITSO-s	ITSO-s	ITSO-s	ITSO-s	ITSO-s	ITSO-s	ITSO-s	ITSO-s	Yes
5	ITSO-p	ITSO-p	ITSO-p	ITSO-p	ITSO-p	ITSO-p	ITSO-p	ITSO-p	Yes
6	ITSO-p	ITSO-p	ITSO-p	ITSO-p	ITSO-p	ITSO-p	ITSO-p	ITSO-p	Yes
7									Yes
8	Prod-a	Prod-a	Prod-a	Prod-a	Prod-a	Prod-a	Prod-a	Prod-a	Yes
9	Prod-a	Prod-a	Prod-a	Prod-a	Prod-a	Prod-a			Yes
A	Prod-b	Prod-b	Prod-b	Prod-b	Prod-b	Prod-b	Prod-b	Prod-b	Yes
B	Prod-b	Prod-b	Prod-b	Prod-b	Prod-b	Prod-b			Yes
C	ATF								Yes
D									Locked
E	LOCK-0	LOCK-1	OTP-0	OTP-1	OTP-2	OTP-3	OTP-4	OTP-5	N/a

Reserved for internal use
Spare OTP bits
Spare

Data Usage for ITSO Small Memory Model Transport Ticket				
Logical Page Number	Blocks of Jewel Memory	Number of Bytes Available	Number of Bytes Required	Data Group Usage
0	0	7	**7**	UID = chip Unique IDentification number
1	1	8	**0**	UIDm = manufacturer Unique Identification number
2	2-4	24	**24**	ITSO-s = ITSO specific data & seal
3	5-7	24	**16**	ITSO-p = ITSO static product data
4	8-9	16	**16**	Prod-a = ITSO dynamic product data & seal – anti-tear-a
5	A-B	16	**16**	Prod-b = ITSO dynamic product data & seal – anti-tear-b
6	C	8	**1**	ATF = Anti-Tear Flag
7	D	Reserved	Reserved	Reserved for internal use
8	E	8	**2**	2 bytes of Block Lock Bits + 6 bytes of OTP Bits
Total		111	**82**	

Notes

The Jewel IC uses a “Digital Certificate” or “Seal” based on the unalterable and unique identification number to authenticate and provide an appropriate level of security.

All blocks and hence all logical pages have an individual one time lock capability

Logical pages 4 and 5 are operated as anti-tear shadows of each other as signalled by the Anti-Tear Flag (ATF)

16. Annex 1 – Hints & Tips

This section is intended to clarify the communication protocol and CRC calculation.

16.1 Command Clarification

The following is an important clarification of Jewel tag commands.

IMPORTANT
• **Commands to a Jewel tag apart from the REQA or WUPA consist of seven bytes followed by a two byte checksum** • **Each byte is sent within its own frame** • **A frame start-of-message sequence proceeds each byte and a frame end-of-message sequence follows each byte**

16.2 REQA Response

The Jewel ATQA response to a REQA or WUPA command consists of byte 0x00 followed by byte 0x0C.

Some RFID documentation specifies the REQA response as a 16-bit value, where the least significant byte is transmitted before the most significant byte. For example, the 16-bit representation of the Jewel ATQA response in this format is 0x0C00 (where 0x00 is transmitted first and 0x0C second).

16.3 Communication Example

The table below illustrates reading and writing to a Jewel tag with UID (Unique IDentification number) = 00,00,00,00 and all memory initialised to zero.

Note that bytes are transmitted with the least significant bit first. For example the table below shows that REQA (0x26, binary 010 0110) is transmitted as logic 0, logic 1, logic 1, logic 0, logic 0, logic 1 and finally logic 0.

Jewel Communication Examples (UID = 00,00,00,00)

In each column, the left-most byte (bit) is transmitted first and the right-most byte (bit) transmitted last. S refers to frame start and E to frame End sequences. The bit sequences are shown in brackets and include the odd parity bit.

Description	Command to tag (Hex)	Response from tag (Hex)
REQA	26 (S 0110010 E)	00,0C (S 00000000 1 00110000 1 E)
RID	78,00,00,00,00,00,00,D0,43 (S 0001111 E S 00000000 E …)	00,00,00,00,00,00,8F,F7
RALL	00,00,00,00,00,00,00,70,8C	00,<120 zero bytes>,00,4D,99 (total 124 bytes)
READ	01,08,00,00,00,00,00,FD,32	08,00,87,C1
WRITE_E	53,08,12,00,00,00,00,41,D5	08,12,14,F2
READ	01,08,00,00,00,00,00,FD,32	08,12,14,F2
RALL	00,00,00,00,00,00,00,70,8C	00,00,00,00,00,00,00,00,00,00,12,00,<109 zero bytes>,00,EA,B3 (total 124 bytes)

16.4 Communication Summary

Commands sent to Jewel tag have the following format:

- First byte is 7 bits
- Remaining bytes are 8 bits
- Least significant bit is sent first
- There are no parity bits
- **A frame start sequence proceeds each byte (including the CRC bytes) and a frame end follows each byte**
- The CRC_B is appended to all commands apart from REQA and WUPA

Responses from the Jewel tag have the following format;

- All bytes are 8 bit
- Least significant bit is sent first
- A parity bit (Odd) follows each byte
- A response begins with a start frame sequence, and ends with a stop frame sequence. The response bytes (including CRC) are between these sequences.
- The CRC_B is appended to all responses apart from ATQA

16.5 CRC Clarification

The CRC is CRC_B as specified by ISO/IEC 14443-3:2001(E) Annex B.

The CRC is always calculated on 8-bit bytes. Although the first Jewel command byte is transmitted as 7-bit, 8-bits must still be used to calculate the CRC (i.e. the 7-bits of the command must be padded with a zero in the MSB). The CRC is calculated on all data bytes, excluding the start, end, parity and CRC bits. The 16-bit CRC is transmitted with the least significant byte first, then the most significant byte.

16.6 Code Sample Written in C for CRC Calculation


```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#define CRC_A 1
#define CRC_B 2
#define BYTE unsigned char

unsigned short UpdateCrc(unsigned char ch, unsigned short *lpwCrc)
{
ch = (ch^(unsigned char)((*lpwCrc) & 0x00FF));
ch = (ch^(ch<<4));
*lpwCrc = (*lpwCrc >> 8)^((unsigned short)ch << 8)^((unsigned short)ch<<3)^((unsigned short)ch>>4);
return(*lpwCrc);
}

void ComputeCrc(int CRCType, char *Data, int Length,
BYTE *TransmitFirst, BYTE *TransmitSecond)
{
unsigned char chBlock;
unsigned short wCrc;
switch(CRCType) {
case CRC_A:
wCrc = 0x6363; /* ITU-V.41 */
break;
case CRC_B:
wCrc = 0xFFFF; /* ISO/IEC 13239 (formerly ISO/IEC 3309) */
break;
default:
return;
}
do {
chBlock = *Data++;
```

```
UpdateCrc(chBlock, &wCrc);
} while (--Length);
if (CRCType == CRC_B)
wCrc = ~wCrc; /* ISO/IEC 13239 (formerly ISO/IEC 3309) */
*TransmitFirst = (BYTE) (wCrc & 0xFF);
*TransmitSecond = (BYTE) ((wCrc >> 8) & 0xFF);
return;
}

BYTE BuffCRC_A[10] = {0x12, 0x34};
BYTE BuffCRC_B[10] = {0x0A, 0x12, 0x34, 0x56};
unsigned short Crc;
BYTE First, Second;
FILE *OutFd;
int i;

int main(void)
{
printf("CRC-16 reference results ISO/IEC 14443-3\n");
printf("Crc-16 G(x) = x^16 + x^12 + x^5 + 1\n\n");
printf("CRC_A of [ ");
for(i=0; i<2; i++) printf("%02X ",BuffCRC_A[i]);
ComputeCrc(CRC_A, BuffCRC_A, 2, &First, &Second);
printf("] Transmitted: %02X then %02X.\n", First, Second);
printf("CRC_B of [ ");
for(i=0; i<4; i++) printf("%02X ",BuffCRC_B[i]);
ComputeCrc(CRC_B, BuffCRC_B, 4, &First, &Second);
printf("] Transmitted: %02X then %02X.\n", First, Second);
return(0);
}
```

16.7 Code Sample Written in Perl for CRC Calculation


```
#!/usr/bin/perl
# CRC calculator for Jewel
# original Aug 2003

print "Reader to tag - Enter hex for each byte, empty string to end\n";
$poly = 0x0810;          #polynomial - we xor bits 11 and 4 with in0
$crc  = 0xffff;  #initial value
$m="0"; $b=0;

for($b=0; $b<200; $b++) {
   print "byte $b: ";
   $_=<STDIN>;
   chomp;
   if ($_ eq "") {last;}
   $m = hex($_);   #data byte
   $v=0x01;        #bit marker - start at LSB of data

   for ($i=0; $i<8; $i++) {
     $crcmsb=($crc & 0x8000)>>15;          #MSB of the crc
     $din = ($m & $v)>>$i;                 #selected data bit
     $in0 = $crcmsb ^ $din;

     if ($in0) {
        $crc = $crc ^ $poly;          #xor with polynomial
     }
     $crc = ($crc << 1) & 0xffff;     #left shift the crc
     if ($in0) {$crc = $crc+1;}

     $v = $v << 1;          #next bit of data
   }
}
#printf "crc before transformation = %x\n", $crc;
#now invert and reflect

$crc = $crc ^ 0xffff;                 #invert
$out = 0;                                     #output value
for ($i=0; $i<16; $i++) {
   $j = 8*int($i/8) + (7-$i%8);       #calculate destination bit (reflection)
   $bit = ($crc>>$i) & 1;
   $out = $out + $bit*(2**$j);
}

printf "output CRC = %x\n", $out;
```

16.8 Scope Traces Illustrating Jewel Communication

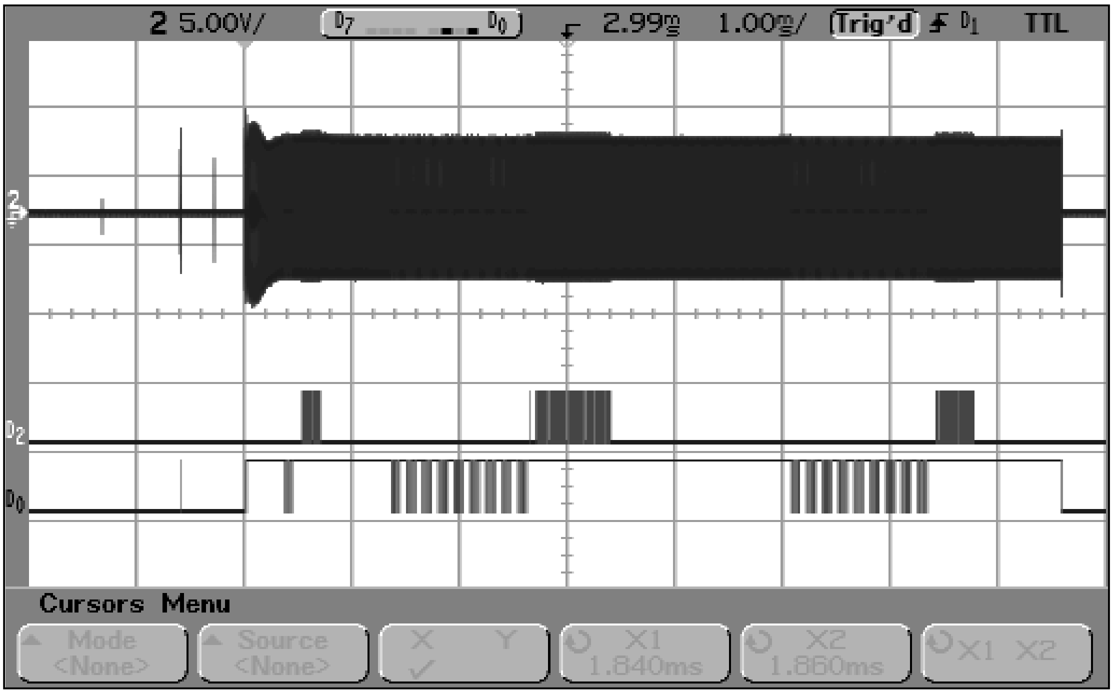
This section uses oscilloscope traces to illustrate Jewel tag communication. The traces show the following signals:

- 2 : Sniffer coil placed next to the reader antenna
- D_0 : Digital signal showing transmissions from the reader to the Jewel Emulator
- D_2 : Digital signal showing transmissions from the Jewel Emulator to the reader

The scope traces were taken during a read of 1 byte from Jewel location 0x08, using the commands REQA, RID and READ. The UID is all zero and the data byte at 0x08 has a value of 0x00. A Jewel tag emulator was used for all the traces.

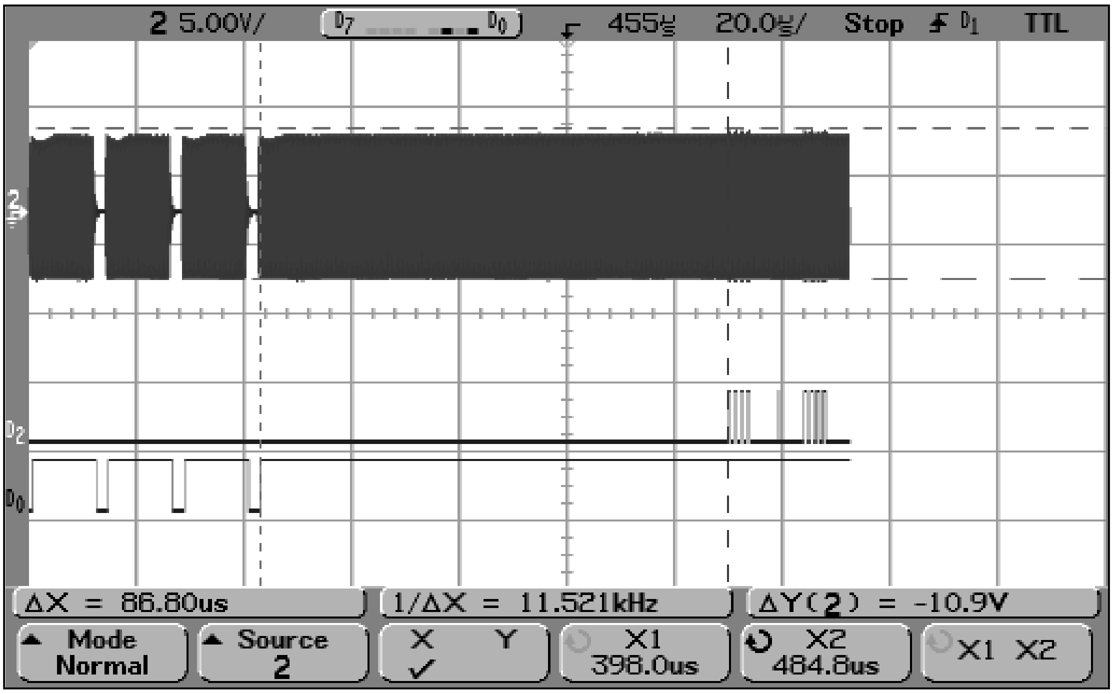
16.8.1 Trace 1: Entire Communication Sequence

Note that the entire communication sequence takes 7.5ms.



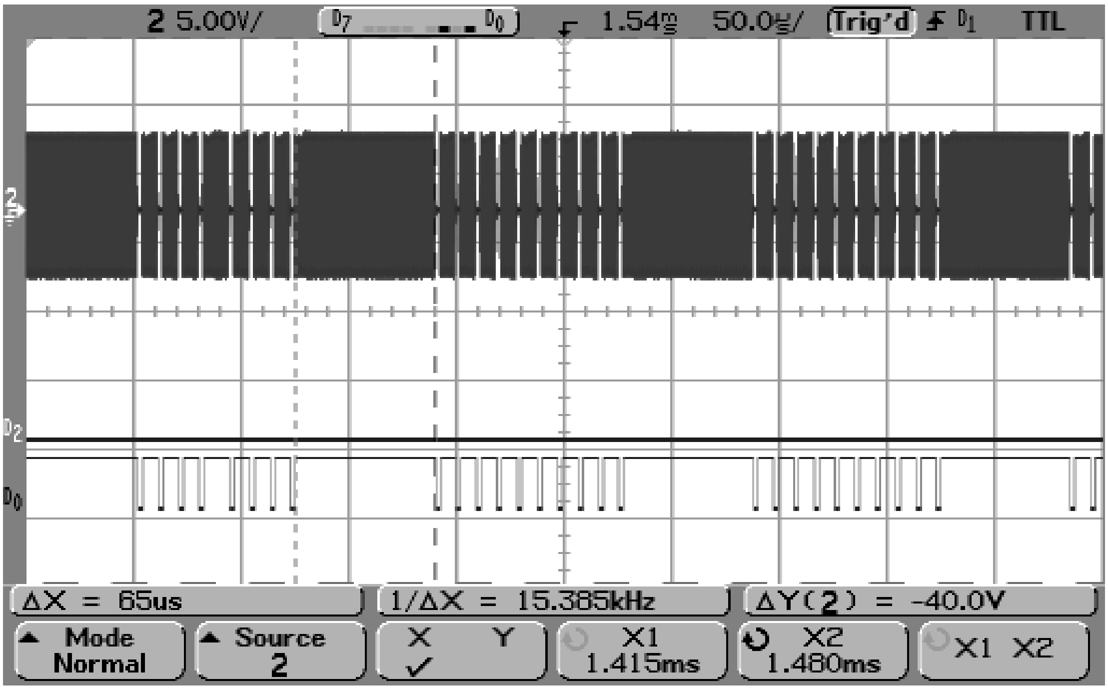
16.8.2 Trace 2: Delay from REQA to ATQA Response

Note that the interval between REQA command to ATQA response is 86.8μs.



16.8.3 Trace 3: Start of RID Command

Note that the inter-frame gap is 65μs.



16.8.4 Trace 4: End of RID Command

Note that the interval between the RID command and the tag response is 87.2µs.